Choosing a Radar Algorithm to Use as a Proxy for E Part 2

Monte Bateman and Douglas Mach

1 More on averaging...

Question: Which algorithm works better — averaging or summing?

Our original premise: Averaging is a bad idea, because it throws away depth information. Frank suggested several "pathological" (non-physical) cases which he claims argues against summing and in favor of averaging. We will show that this is *not* the case.

A large part of Frank's cases were based on the "arithmetic" problems that arise from negative and zero values of reflectivity. The approach we used was to eliminate/ignore values ≤ 0 because they won't contribute to electrification. However, in the interests of completeness, we go through these cases and show that "mathematically" it's not a problem.

1.1 Case 1: A single box of 0 dBZ reflectivity

0

In this case, the depth is 1, so there is no depth information to be lost. We get the following:

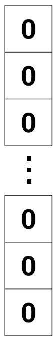
sum	0
mean	0
offset sum	offset value
offset mean	offset value

where "offset" means to add a constant to the reflectivity data values. The reason for this is so that a value of 0 dBZ is different from the value for "missing data." It also makes the data unipolar. Typically, the constant is chosen to be equal to the smallest possible value, say 10 or 20 dBZ if the minimum reflectivity value is -10 or -20 dBZ. Also note that the minimum detectable signal goes up with range — which helps!

In this case both the sum and the mean report the same thing. Not surprisingly, since there is no depth information to lose. So the score is (using the offsetting technique, where appropriate):

sum	1
mean	1

1.2 Case 2: Multiple (n) boxes of 0 dBZ reflectivity



Here, we have depth information. We have n boxes of 0 dBZ reflectivity, and n is the depth of the cloud. This case gives the following results:

sum	0
mean	0
offset sum	depth $\times$ offset value
offset mean	offset value

If we use the offset technique, the sum detects the depth of cloud. The average gives the same result as Case 1. In the case, the (offset) sum wins.

So now the score is:

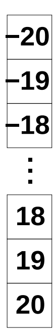
sum	2
mean	1

1.3 Case 3: A deep cloud that smoothly progress from -20 to 20 dBZ reflectivity

Here we have a deep cloud that linearly progresses from 20 to -20 dBZ. This also gives a sum and mean of 0. If we offset by 20 dBZ, so that the data now range from 40 to 0, we get the following:

sum	0
mean	0
offset sum	depth $\times$ offset value
offset mean	offset value

Again, using the offset technique, the sum detects the depth of cloud. The average gives the same result as in Cases 1 & 2! Again, the (offset) sum wins, and the score is:

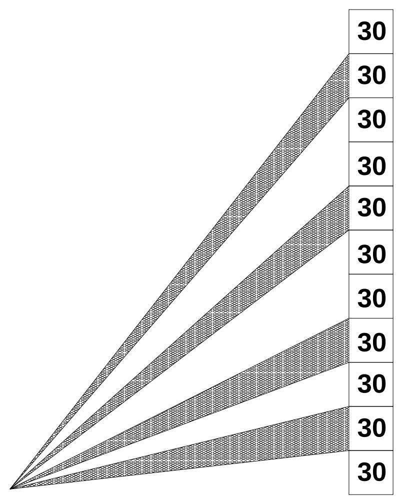


sum	3
mean	1

2 Which method better handles scan gaps in the radar data?

We will examine two cases: a uniform cloud, and a very non-uniform cloud. Note that we have been talking about gridded data, in which the radar scans gaps have been filled by interpolation. Here, we still use the "boxes" to make our point, but we realize that "scan gaps" and gridded data are mutually exclusive.

2.1 Uniform cloud

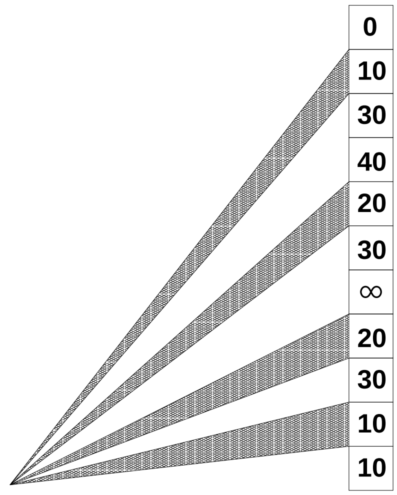


In this case, the cloud is a 11 km deep with a uniform, 30 dBZ reflectivity. The radar scans 4 of these levels and misses 7. So the results would be:

sum	120
mean	30

Again, the average is throwing out the depth information, making the sum a more useful tool for assessing the state of the cloud.

2.2 A very non-uniform cloud



In this case, the cloud is a 11 km deep with highly erratic reflectivity values. Again, the radar scans 4 of these levels and misses 7.

sum	60
mean	15

Here we see the problem. Neither the average nor the sum accurately characterizes this cloud! Missed data are simply that: missed data.

Trying to choose an algorithm that handles scan gaps is trying to answer the wrong question.

3 Conclusion

In each of the specific pathological cases in section 1, the (offset) mean gives the same answer — it has no skill at separating these clouds. In each case, the (offset) sum shows that it is able to differentiate between these clouds.

Secondly, arguing in favor or against an algorithm regarding scan gaps, is trying to answer the wrong question. It's a bit like asking if I can make my car go faster by adding pinstripes or by painting it blue.

The point is that *any* radar-based algorithm will suffer because of scan gaps.